An empirical comparative study of approximate methods for binary graphical models; application to the search of associations among causes of death in French death certificates.

Vivian Viallon , Onureena Banerjee
Grégoire Rey , Eric Jougla
Joël Coste

Abstract

Looking for associations among multiple variables is a topical issue in statistics due to the increasing amount of data encountered in biology, medicine and many other domains involving statistical applications. Graphical models have recently gained popularity for this purpose in the statistical literature. Following the ideas of the LASSO procedure designed for the linear regression framework, recent developments dealing with graphical model selection have been based on ℓ_1-penalization. In the binary case, however, exact inference is generally very slow or even intractable because of the form of the so-called log-partition function. Various approximate methods have recently been proposed in the literature and the main objective of this paper is to compare them. Through an extensive simulation study, we show that a simple modification of a method relying on a Gaussian approximation achieves good performance and is very fast. We present a real application in which we search for associations among causes of death recorded on French death certificates.

1 Introduction

In biology, medicine, and many other domains of statistical application, researchers are increasingly faced with problems involving numerous variables and a natural problem is that of studying their relationships. Standard examples are the construction of social or communication networks and systems biology. When the underlying variables are binary (which is the focus of this paper), a classical way for studying their relationships is to use *Poisson log-linear models* for multiway *contingency tables* (Agresti, 1990)(Mccullagh and Nelder, 1989). How to perform selection in log-linear models, or equivalently in binary graphical models, depends upon the number of variables p. Indeed, the total number of cell entries for a p-way contingency table is 2^p, and the total number of free parameters in the associated saturated model is $2^p - 1$. When p is low, a standard approach for model selection is greedy stepwise forward-selection or backward-deletion: in each step, selection or deletion is based on hypothesis testing at some level α. However, the computational complexity even for modestly dimensioned contingency tables plus the multiple hypothesis testing issues related to such a procedure has made it unpopular in this context. Consequently, Dahinden et al. (2007)

recently performed selection in log-linear models by using an ℓ_1-penalized version of the log-likelihood, extending the LASSO ideas (Tibshirani, 1996) originally designed for selection in linear models. However, computing the (penalized) log-likelihood for log-linear models generally requires the enumeration of each of the 2^p profiles, which is not plausible for large p (e.g., larger than about 30). For such moderate-to-large values of p, alternative methods are required.

Roughly speaking, two approaches have been proposed in the literature. First, exact inference can be performed in the case of highly sparse models. For instance, exact computation via the junction tree algorithm is manageable for highly sparse graphs but becomes unwieldy for dense graphs (Lee et al., 2007). The second approach is to use approximate inference. Notably, much attention has recently been paid to methods relying on proxies for the exact likelihood. Höfling and Tibshirani (2009) and Wang et al. (2010) proposed two distinct algorithms to maximize an ℓ_1-penalized version of the so-called *pseudo-likelihood* (Besag, 1975). These methods are closely related to the one formerly proposed by Wainwright et al. (2006) who used ℓ_1-penalized logistic regressions on each single node to construct the whole graphical model. Besides these three methods, Banerjee et al. (2008) used a convex relaxation technique to derive a Gaussian approximate log-likelihood as well as its sparse maximum solution.

Interestingly, Höfling and Tibshirani (2009) showed through an extensive simulation study that approximate solutions (either solutions maximizing the pseudo-likelihood or those derived from the method proposed by Wainwright et al. (2006)) are much faster and only slightly less accurate than exact methods. However, no empirical evaluation of the Gaussian approximate solution proposed by Banerjee et al. (2008) has ever been conducted and filling this gap is the primary objective of this paper. We thereby propose to conduct such an evaluation by comparing this method with the two other approximate methods of Wainwright et al. (2006) and Höfling and Tibshirani (2009) on simulated data.

Here is a brief outline of the paper. In Section 2 we first summarize the principles of the aforementioned approximate methods. We then present results from an extensive empirical comparison study. A slight modification of the method proposed by Banerjee et al. (2008) is especially shown to achieve very good accuracy and to be extremely fast. Finally, we present an application from a real example where we looked for associations among causes of death in the database of French death certificates of the year 2005 (Section 4).

2 Approximate methods for binary graphical models

2.1 The Ising model

Let $\mathbf{X} = (X^{(1)}, ..., X^{(p)}) \in \{0,1\}^p$ be a p-dimensional vector of binary random variables. Given a random sample $\mathbf{X}_1, ..., \mathbf{X}_n$ of i.i.d. replicae of $\mathbf{X}$, we wish to study the associations between the coordinates of $\mathbf{X}$. One way to do so is to construct the binary graphical model for the random vector $\mathbf{X}$, that is an undirected graph $\mathcal{G} = (V, E)$, where V contains p vertices corresponding to the p coordinates and the edges $E = (e_{k,\ell})_{1 \leq k < \ell \leq p}$ describe the conditional independence relationships among $X^{(1)}, ..., X^{(p)}$. The edge $e_{k,\ell}$ between $X^{(k)}$ and $X^{(\ell)}$ is absent if and only if $X^{(k)}$ and $X^{(\ell)}$ are independent conditional on the other variables. For binary graphical models, it is common to focus only on the family of probability distributions given by the quadratic exponential binary model (Wainwright et al., 2006; Banerjee et al., 2008;

Höfling and Tibshirani, 2009; Wang et al., 2010), also known as the Ising model. Namely, for all $\mathbf{x} = (x^{(1)}, ..., x^{(p)}) \in \{0,1\}^p$, we assume that the probability of $\mathbf{x}$ is given by

$$P(\mathbf{x}, \Theta) = \exp\Big\{ \sum_{k=1}^{p} \theta_k x^{(k)} + \sum_{k=1}^{p-1} \sum_{\ell=k+1}^{p} \theta_{k,\ell} x^{(k)} x^{(\ell)} - A(\Theta) \Big\}, \tag{1}$$

where the so-called log *partition function* $A(\cdot)$ is defined as follows

$$A(\Theta) = \log \sum_{\mathbf{x} \in \{0,1\}^p} \exp\Big\{ \sum_{k=1}^{p} \theta_k x^{(k)} + \sum_{k=1}^{p-1} \sum_{\ell=k+1}^{p} \theta_{k,\ell} x^{(k)} x^{(\ell)} \Big\}. \tag{2}$$

Note that $A(\cdot)$ is strictly convex and ensures that $\sum_{\mathbf{x} \in \{0,1\}^p} P(\mathbf{x}, \Theta) = 1$ (note also that the strict convexity of this function ensures the identifiability of the parameter matrix Θ). From (1), we have

$$\frac{P(X^{(k)} = 1, X^{(\ell)} = 1 | X^{(j)}, j \neq k, \ell) / P(X^{(k)} = 0, X^{(\ell)} = 1 | X^{(j)}, j \neq k, \ell)}{P(X^{(k)} = 1, X^{(\ell)} = 0 | X^{(j)}, j \neq k, \ell) / P(X^{(k)} = 0, X^{(\ell)} = 0 | X^{(j)}, j \neq k, \ell)} = \exp(\theta_{k,\ell}). \tag{3}$$

Therefore, the parameters $\theta_{k,\ell}$ are the conditional log-odds ratios. The conditional independence between $X^{(k)}$ and $X^{(\ell)}$ is then equivalent to $\theta_{k,\ell} = 0$, that is, the edge $e_{k,\ell}$ is absent if and only if $\theta_{k,\ell} = 0$. Consequently, selection in binary graphical models is equivalent to identifying the (k, ℓ) pairs for which $\theta_{k,\ell} = 0$.

From (1), using the fact that $x^{(k)}$ is binary, and denoting by $\mathcal{X} = (\mathbf{X}_1, ..., \mathbf{X}_n)^T$ the matrix representing the whole dataset, the ℓ_1-penalized log-likelihood writes

$$l(\mathcal{X}, \Theta) = \sum_{\ell \geq k \geq 1}^{p} (\mathcal{X}^T \mathcal{X})_{k,\ell} \theta_{k,\ell} - nA(\Theta) - n\lambda \|\Theta\|_1, \tag{4}$$

where Θ is a symmetric matrix with $\theta_{kk} = \theta_k$ for $k = 1, ..., p$. However, because of the complexity of the log-partition function, methods based on approximate inference are needed in most cases and we recall the principle of three of them in the following paragraphs.

Let us begin by recalling that the approximation established by Banerjee et al. (2008) is only valid for the first-order-interaction log-linear model described above. On the other hand, Wainwright et al. (2006), Höfling and Tibshirani (2009) and Wang et al. (2010) also only consider this simple model but higher-order interaction models can be (at least theoretically) handled with these methods, at a cost of a dramatically increased computational time.

2.1.1 Multiple logistic regressions

From (1), it is easy to see that, for all $k = 1, ..., p$, setting $\mathbf{x}^{(-k)} = (x^{(1)}, ..., x^{(k-1)}, x^{(k+1)}, ..., x^{(p)})$, we have

$$\text{logit}\{\mathbb{P}(X^{(k)} = 1 | \mathbf{x}^{(-k)})\} = \sum_{\ell \neq k} \theta_{k,\ell} x^{(\ell)} + \theta_k. \tag{5}$$

Wainwright et al. (2006) then extensively study a method (which will be referred hereafter as `SepLogit` following the terminology adopted in Wang et al. (2010)) in which ℓ_1-penalized logistic regression is used to estimate the neighborhood of each of the p nodes in the graph

separately. Wainwright et al. (2006) gives rigorous consistency results in a high-dimensional setting, where the number of nodes is allowed to grow as a function of the number of samples. The authors give sufficient conditions under which the method will consistently estimate the neighborhood for every node in the graph simultaneously. In a sense, the paper can be seen as a discrete version of Meinshausen and Bühlmann (2006).

For a finite number of samples, the p logistic regression problems are solved separately and, since the results may be asymmetric, they can be combined in one of two ways to draw a graph. One possibility is to draw an edge between two nodes in the graph only if each node is estimated to belong to the neighborhood of the other (method `SepLogit AND`). Alternatively, we can decide to draw an edge between two nodes so long as at least one of them is estimated to belong to the neighborhood of the other (method `SepLogit OR`).

In our empirical comparison study, this method will be implemented using the coordinate descent procedure developed by Friedman et al. (2008b) (and implemented in the `glmnet` R package).

2.1.2 Pseudo-likelihood maximization

One of the shortcomings of the method proposed by Wainwright et al. (2006) is the aforementioned asymmetry. To overcome this limitation, Höfling and Tibshirani (2009) and Wang et al. (2010) recently proposed to use the pseudo-(log-)likelihood, first suggested by Besag (1975). The pseudo-likelihood is formally defined as

$$\sum_{i=1}^{n}\sum_{k=1}^{p}\log\{\mathbb{P}(X_i^{(k)}|X_i^{(1)},...X_i^{(k-1)},X_i^{(k+1)},...,X_i^{(p)})\}. \tag{6}$$

Accordingly, the approach based on the maximization of the ℓ_1-penalized pseudo-likelihood solves all p logistic regression problems simultaneously, while enforcing symmetry. Apart from symmetry enforcement, this methods differs from the one studied in Wainwright et al. (2006) in that the ℓ_1-norm penalty is applied to the entire network, while in `SepLogit` it is applied to each neighborhood.

For future use, and still denoting by $\mathcal{X} = (\mathbf{X}_1, ..., \mathbf{X}_n)^T$ the matrix representing the whole dataset, observe that the pseudo-likelihood writes

$$\text{pseudo-}l(\mathcal{X},\Theta) = -\sum_{i=1}^{n}\sum_{k=1}^{p}\log\left\{1+\exp(-\tilde{x}_i^{(k)}\mathcal{X}^k[i,]\Theta[,k])\right\}, \tag{7}$$

where $\mathcal{X}^k$ is the same as $\mathcal{X}$ with kth column set to 1 and $\tilde{x}_i^{(k)} = 2x_i^{(k)} - 1$ (i.e., $\tilde{x}_i^{(k)}$ is the spin version of $x_i^{(k)}$). Here and elsewhere, $M[i,]$ (resp. $M[,k]$) denotes the i-th row (resp. k-th column) of a matrix M.

Höfling and Tibshirani (2009) first develop and implement an algorithm for maximizing the ℓ_1-penalized pseudo-likelihood function, using a local quadratic approximation to the pseudo-likelihood. They then use this algorithm as a building block for a new algorithm that maximizes the true log-likelihood. However, as we already said, they observed that the approximate pseudo-likelihood is much faster than the exact procedure, and only slightly less accurate. Therefore, to save computational time, we only considered the approximate pseudo-likelihood

in this paper. In the forthcoming empirical comparison study, this method will be implemented using the `BMN` R package and will be referred to as `BMNPseudo`.

Interestingly, and as pointed out by Höfling and Tibshirani (2009), the derivative of the pseudo-likelihood on the off-diagonal is roughly twice as large as the derivative of the exact likelihood. Moreover, in the case $p = 2$, it is easy to see that the deviance of the model with no association (i.e. minus twice the difference between the log-likelihood of this model and the log-likelihood of the saturated model) when computing with the pseudo-likelihood is twice as large as the one computed with the exact likelihood (while, obviously, the pseudo-likelihood coincides with the exact likelihood for the model with no interaction). The generalization of this striking result for higher p is not straightforward, but our empirical examples suggest it may hold at least approximately (see Section 3.3). Therefore, we will consider methods relying on both pseudo-$l(\mathcal{X}, \Theta)$ (method `BMNPseudo`) and pseudo-$l(\mathcal{X}, \Theta)/2$ (method `BMNPseudo 1/2`) in the sequel.

For the sake of completeness, we shall add that Wang et al. (2010) develop a gradient-descent algorithm to maximize the ℓ_1-penalized pseudo-likelihood. They further propose an extension to account for spatial correlation among the variables (which was relevant in their example dealing with genomic data). However, the corresponding `LogitNet` R package was not available at the time we wrote this paper, so we were not able to include it in our empirical comparison study.

2.1.3 Gaussian Approximation of the Ising log-likelihood

The basic idea of the method described by Banerjee et al. (2008) is to replace the log-partition function in the Ising model with an upper bound suggested by Wainwright and Jordan (2006). The resulting approximation can then, with some manipulation, be put in a form that can be solved efficiently using block coordinate descent. In order to add some specific details, we shall define some notation. Denote by $(\mathbf{Z}_1, ..., \mathbf{Z}_n) \in \{-1, 1\}^{p \times n}$ the spin version of $(\mathbf{X}_1, ..., \mathbf{X}_n)$, and let $\overline{Z}^{(k)}$ denote the sample mean of variable $Z^{(k)}$, for $k = 1, ..., p$. Now, define the empirical covariance matrix S as

$$S = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{Z}_i - \overline{Z})(\mathbf{Z}_i - \overline{Z})^T, \tag{8}$$

where $\overline{Z}$ is the vector of sample means $\overline{Z}^{(k)}$. Making use of a *convex relaxation* and a useful upper bound on the log-partition function obtained by Wainwright and Jordan (2006), Banerjee et al. (2008) established that an approximate sparse maximum likelihood solution for a given λ has the following form

$$\begin{aligned} \hat{\theta}^{\lambda}_{k} &= \overline{Z}^{(k)}, \\ \hat{\theta}^{\lambda}_{k,\ell} &= -(\hat{\Sigma}^{-1}_{\lambda})_{k\ell}, \end{aligned} \tag{9}$$

where the matrix $\hat{\Sigma}^{-1}_{\lambda}$ is the solution of the following optimization problem

$$\hat{\Sigma}^{-1}_{\lambda} = \arg\max_{M} \{ \log |M| - \mathrm{tr}(M(S + \mathrm{diag}(1/3))) - \lambda \|M\|_1 \}. \tag{10}$$

More precisely, Banerjee et al. (2008) proposed a block-coordinate descent algorithm to solve a dual formulation of (10), which can be written as

$$\hat{\Sigma}_\lambda = \arg\max_W \{ \log|W| : W_{kk} = S_{kk} + \frac{1}{3}, |W_{k\ell} - S_{k\ell}| \leq \lambda \}. \tag{11}$$

In the Gaussian case, Banerjee et al. (2008) showed that the ℓ_1-penalized covariance selection problem could be written

$$\widehat{\Sigma} = \arg\max_W \{ \log|W| : \|W - S_G\|_\infty \leq \lambda \}, \tag{12}$$

where S_G is the empirical covariance matrix attached to a given sample of Gaussian vectors. An algorithm for handling binary graphical models can be derived by comparing (11) and (12). The original $\{0,1\}$ data has first to be transformed into $\{-1,1\}$ data. Then, adding the constant $1/3$ to the diagonal elements of the resulting empirical covariance matrix, the algorithms developed in the Gaussian case (in particular the `glasso` R package developed by Friedman et al. (2008a)) can be reused.

A common question when working with Gaussian variables is whether to standardize them, or equivalently, whether to use the covariance or the correlation matrix. Moreover, in the binary case, the correlation coefficient between two variables (also known as the ϕ-coefficient) is closely related to the χ^2 statistic used to test for (marginal) independence. Putting these two observations together, we decided to evaluate a simple modification of the method proposed by Banerjee et al. (2008) where the quantity S+diag$(1/3)$ is replaced by the correlation matrix. Lastly, we also decided to evaluate the modification in which S+diag$(1/3)$ is simply replaced by S.

To recap, we will consider the three following optimization problems

$$\widehat{C}^\nu_\lambda = \arg\max_M \{ \log|M| - \mathrm{tr}(MS^\nu) - \lambda\|M\|_1 \}, \text{ for } \nu = 1,2,3, \tag{13}$$

where $S^1 = (\mathrm{Cov}(\mathbf{Z}) + \mathrm{diag}(1/3))$, $S^2 = \mathrm{Cov}(\mathbf{Z})$ and $S^3 = \mathrm{Cor}(\mathbf{Z})$. For any λ, and every $\nu = 1,2,3$ an estimation of $\theta_{k,\ell}$ is then given by $-(\widehat{C}^\nu_\lambda)_{k\ell}$.
In our empirical comparison study, the three methods will be implemented using the `glasso` R package (Friedman et al., 2008a) and will be referred to as `GaussCov 1/3`, `GaussCov` and `GaussCor` for the choices $\nu = 1$, $\nu = 2$ and $\nu = 3$ respectively (we may as well use the generic expression `GaussApprox` when dealing with either methods).

2.2 Sparsity parameter selection

Two procedures for selecting tuning parameters are generally considered, namely cross-validation (CV) and Bayesian Information Criterion (BIC), the latter being computationally more efficient as suggested by Yuan and Lin (2007) for instance. In the case of Gaussian graphical models, Gao et al. (2009) further demonstrate the advantageous performance of BIC for sparsity parameter selection through simulation studies. In this paper, we therefore decided to only consider BIC.

When trying to select the optimal sparsity parameter λ using either CV or BIC, however, one has to pay attention to the following fact. Since, for each $\lambda > 0$, estimates of the parameters

of interest are shrunk, using them for choosing λ from CV or BIC often results in severe over-fitting (Efron et al., 2004). Therefore, un-shrunk estimates have to be derived before computing the BIC.

Taking the example of methods `GaussApprox`, for any λ, we have to compute the un-shrunk matrix

$$\widetilde{C}^{\nu}_{\lambda} = \arg\max_{M \in \mathcal{M}^{+}_{\lambda}} \{ \log |M| - \mathrm{tr}(MS^{\nu}) \}. \tag{14}$$

Here, $\mathcal{M}^{+}_{\lambda} = \{M \succ 0 : M_{k,\ell} = 0 \text{ for couples } (k,\ell) \text{ such that } (\widehat{C}^{\nu}_{\lambda})_{k,\ell} = 0\}$ ($\{M \succ 0\}$ being the set of positive definite matrices of order p, and $\widehat{C}^{\nu}_{\lambda}$ being as in (9)). To solve the optimization problem (14), one approach is to reuse the algorithm used to solve (13) after replacing the scalar parameter λ by the penalty matrix Λ such that $\Lambda_{k,\ell} = 0$ if $\hat{\theta}^{\lambda}_{k,\ell} \neq 0$, and $\Lambda_{k,\ell} = \infty$ otherwise, where $\hat{\theta}^{\lambda}_{k,\ell}$ is the shrunk estimation of the coefficient $\theta_{k,\ell}$ obtained with the value λ and is used as an initial value for the optimization. Alternative approaches might be considered, such as the algorithm developed by Dahl et al. (2008) for instance.

Defining

$$\mathcal{L}^{\nu}_{\lambda} = \log |\widetilde{C}^{\nu}_{\lambda}| - \mathrm{tr}(\widetilde{C}^{\nu}_{\lambda} S^{\nu}),$$

the BIC procedure now simply corresponds to selecting the sparsity parameter $\lambda^{\nu}_{\mathrm{BIC}}$ such that

$$\lambda^{\nu}_{\mathrm{BIC}} = \arg\max_{\lambda} \{ n\mathcal{L}^{\nu}_{\lambda} - K^{\nu}_{\lambda} \log(n) \}, \tag{15}$$

where $K^{\nu}_{\lambda} = \sum_{k \geq \ell} \mathbb{1}\{(\widehat{C}^{\nu}_{\lambda})_{k,\ell} \neq 0\}$ is the degree of freedom of the model selected with the sparsity parameter λ (Yuan and Lin, 2007).

2.3 Estimation of the conditional odds-ratios

In the binary case, a standard measure of the strength of association between two variables is the (conditional) odds-ratio, which is related to coefficient $\theta_{k,\ell}$ (see (3)). Therefore, consistent estimates of the parameters $\theta_{k,\ell}$ would yield consistent estimates of the conditional odds-ratios. Here again un-shrunk estimates are preferable, and the methods described in the previous section have to be used.

3 Simulation study

In this section, we compare the model selection performances as well as the computational time for the methods described in the previous Section. Results are presented for $p = 10$ and $p = 50$. The choice $p = 10$ has been made for several reasons. First, for such low values of p the true log-likelihood can be quickly computed, and we can then compare it with the approximate log-likelihoods (see Section 3.3 below). Second, approximate methods are still faster to compute when p is small, and conclusions drawn in the case of low p are likely to hold for high p as well (as will be confirmed from our results).

3.1 Evaluation criteria

For each method, every value of λ corresponds to a sparsity structure for the matrix Θ that can be compared with the true sparsity structure. Namely, for all λ and for each method,

we can compute the rate of true positives (correctly identified associations), the rate of false positives (incorrectly identified associations) as well as the overall accuracy. Precision and recall (the latter being identical to the true positive rate) can also be computed, as well as their harmonic mean, often referred to as the F1-score.

In a first evaluation study, we present for each method the performances achieved by the "oracle" model, that is the model constructed with optimal sparsity parameter regarding accuracy. Such an evaluation was not conducted for `SepLogit` because under this method the ℓ_1 penalty is applied to each neighborhood and the "oracle" model would invariably coincide with the true model. The alternative would be to force the algorithm to choose the same sparsity parameter value for every regression model. However, using this alternative approach, we sometimes obtained "oracle" models that achieved performances worse than the models selected by the BIC procedure. Therefore, we do not recommend to force the algorithm to choose the same sparsity parameter value for every regression model.

In a second evaluation study, we present for each method the performances achieved by the model selected according the BIC procedure described above. For methods `GaussApprox`, un-shrunk estimates were derived along the lines described in Section 2.2. A similar approach was used for method `BMNPseudo`. The `BMN` R package also allows the use of a matrix of penalty coefficients. For `SepLogit`, we had to slightly adapt this approach because the `glmnet` package does not allow for building models with only un-penalized coefficients. So, whenever needed, a standard logistic regression model was used to get un-shrunk estimates. This may make the method a little slower, but not much since for each variable, this situation can only arise for the smallest tested λ value, and only if this smallest tested λ value corresponds to the saturated model.

In addition, the computational time is reported. More precisely, we used a grid $[\lambda^{\min} := \lambda^{\max}/1000, ..., \lambda^{\max}]$ of 50 equally-spaced values (on a log-scale) for the parameter λ and we report the time needed to compute the 50 corresponding models for each method ($\lambda^{\max}$ is the data derived smallest value for which all coefficients are zero). Each method was run on a Windows Vista machine with Intel Core 2DUO 2.26GHz with 4GB RAM in the case $p = 10$ and on a MAC Pro machine with intel Xeon 2×2.26GHz Quad Core with 6GB RAM in the case $p = 50$ (the MAC Pro machine was approximately 3.5 times as fast as the Windows machine).

3.2 Data generation

3.2.1 The case $p = 10$

In model (1), given that n individuals are observed, the distribution of the corresponding cell counts $\mathbf{n} = (n_{\mathbf{x}}, \mathbf{x} \in \{-1, 1\}^p)$ is multinomial with probability $\mathbf{P} = (P(\mathbf{x}, \Theta), \mathbf{x} \in \{-1, 1\}^p)$. Accordingly, given a symmetric matrix Θ, data were drawn from the multinomial distribution with probability vector $\mathbf{P}$. Four matrices $\Theta^{(1)}$, $\Theta^{(2)}$, $\Theta^{(3)}$ and $\Theta^{(4)}$ were considered, leading to four different simulation designs.
For $\Theta^{(1)}$, "primary" coefficients $\theta_{k,\ell}$ were simulated independently using a normal distribution with mean zero and variance σ for some $\sigma > 0$. Subsequently, only coefficients $\theta_{k,\ell}$ with an absolute value greater than 0.06 (corresponding to a conditional odds-ratio of $\exp(4 \times 0.06) \simeq 1.27$, since for $\{-1, 1\}$ variables, the conditional odds-ratio is $\exp(4\theta_{k,\ell})$) were retained, all

others being set to 0. The function $A(\Theta^{(1)})$ was then computed according to Equation (2). Selecting $\sigma = 0.05$ led to a true model with 10 associations (among the $p(p-1)/2 = 45$ potential associations).

Matrices $\Theta^{(2)}$ and $\Theta^{(3)}$ were constructed so that they share the same sparsity pattern as $\Theta^{(1)}$, i.e.,

$$\{(k,\ell) : \theta^{(1)}_{k,\ell} = 0\} = \{(k,\ell) : \theta^{(2)}_{k,\ell} = 0\} = \{(k,\ell) : \theta^{(3)}_{k,\ell} = 0\}$$

but they have different non-zero coefficients. In either cases, we selected $(\theta_{1,1}, ..., \theta_{10,10}) = (-1.3, ..., 0)$. For matrix $\Theta^{(2)}$, the non-zero $\theta_{k,\ell}$ coefficients were set to ± 0.2, while they were set to ± 0.4 for matrix $\Theta^{(3)}$.

For matrix $\Theta^{(4)}$, we proceeded as for matrix $\Theta^{(1)}$ but we selected $\sigma = 0.3$ and only the $\theta_{k,\ell}$ coefficients with an absolute value greater than 0.2 were retained (the others being set to 0). Moreover, we selected $(\theta_{1,1}, ..., \theta_{10,10}) = (-1.8, ..., 0)$. This led to a true model with 19 associations.

A graphical representation of matrices $\Theta^{(1)}$, $\Theta^{(2)}$, $\Theta^{(3)}$ and $\Theta^{(4)}$ as well as the corresponding marginal probabilities $\mathbb{P}(X^{(k)} = 1)$, for $k = 1, ..., 10$, estimated on a sample of size $n = 2500$ are presented on Figure 1.

3.2.2 The case $p = 50$

For $p = 50$, we first considered the case of block-diagonal matrices Θ. For $j = 1, 2, 3, 4$, we then used matrices Θ of the form $\text{diag}(\Theta^j, \Theta^j, \Theta^j, \Theta^j, \Theta^j)$.

In a fifth example, matrix $\Theta^{(5)}$ was build as follows. For every $k > \ell \geq 1$, we first draw one observation u from a (0,1)-uniform distribution, and $\theta^{(5)}_{k,\ell}$ was then set to 0 if $u < 0.9$, $\log(2)$ if $u \geq 0.95$ and $\log(1.5)$ otherwise. The resulting true model consisted of 125 edges. Coefficients $\theta^{(5)}_{k,k}$ were set to (logit(0.1),...,logit(0.2)). Gibbs sampling was further used to generate the data (consisting of $\{0, 1\}$ variables this time).

3.3 Empirical comparison of the approximate deviances

In this section, our goal is to empirically evaluate the approximate likelihoods on which the methods under study rely. To do so, we will focus on the case where $p = 10$ since the exact log-likelihood of the Poisson log-linear model can be computed for such a value of p. For each of the four Θ matrices described above, we proceed as follows. We generate a random sample of size $n = 500$, and for each value of the tuning parameter λ on an appropriate grid, we apply method `BMNPseudo`. This leads to some sparsity structure in the corresponding Ising model and we can then compute the Gaussian approximate log-likelihoods $\mathcal{L}^{\text{G}_\nu}_\lambda$, $\nu = 1, 2, 3$ (see (18) below) as well as both the exact Poisson log-linear log-likelihood and the pseudo-likelihood for the Ising model corresponding to this particular sparsity structure. More precisely, the following quantities were considered,

$$\mathcal{L}^{\text{Ps}}_\lambda = -\sum_{i=1}^{n}\sum_{k=1}^{p} \log\left\{1 + \exp(-\tilde{x}^{(k)}_i \mathcal{X}^k[i,]\Theta_\lambda[,k])\right\} \tag{16}$$

$$\mathcal{L}^{\text{Po}}_\lambda = \sum_{i=1}^{n} \log\left\{P(\mathbf{x}_i, \Theta^{\text{Po}}_\lambda)\right\} \tag{17}$$

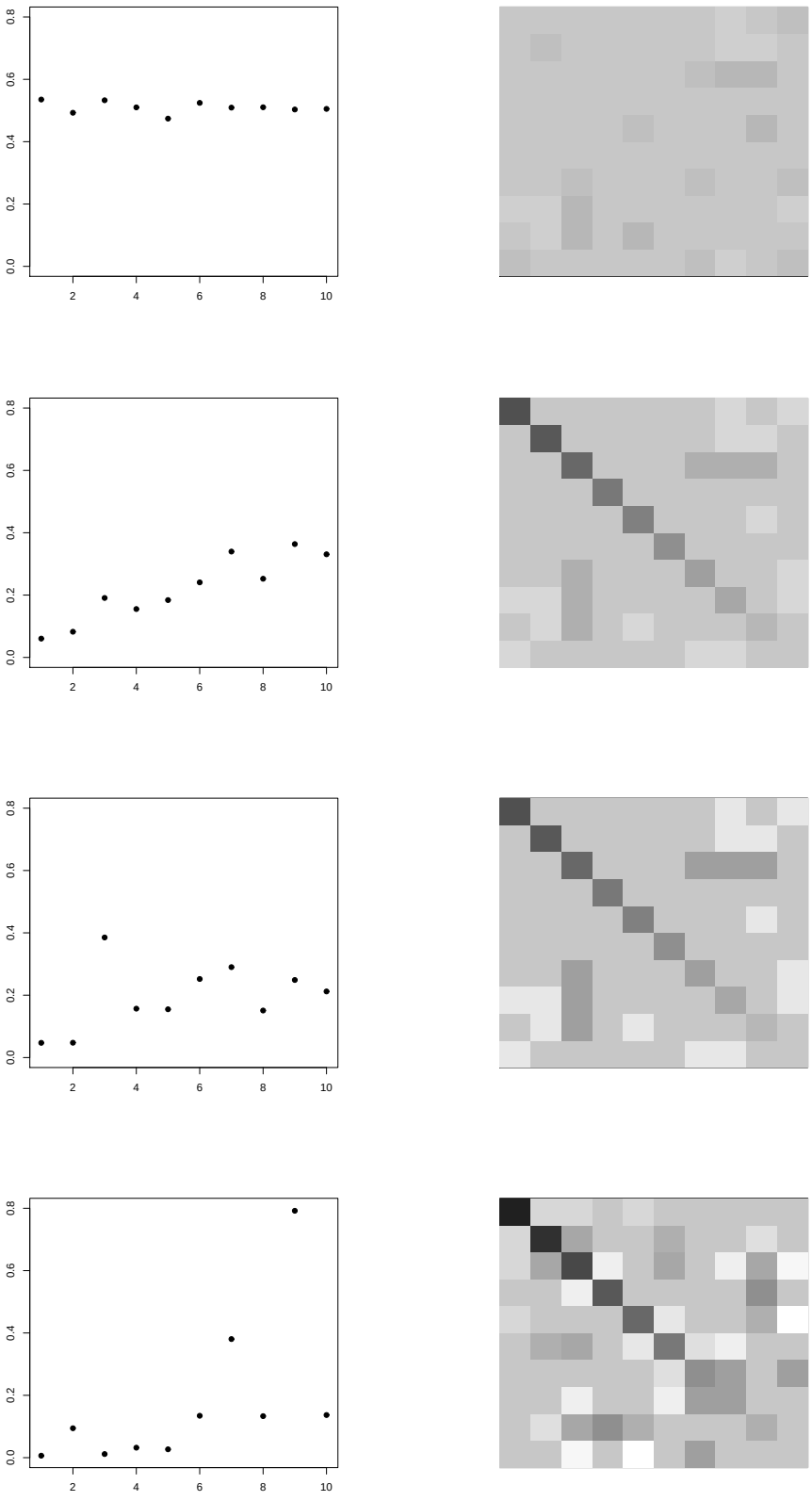

Figure 1: Marginal probabilities $\mathbb{P}(X^{(k)} = 1)$, for $k = 1, ..., 10$, estimated on a sample of size $n = 2500$ generated using the matrix $\Theta = \Theta^{(j)}, j = 1, 2, 3, 4$ are presented in the left panels. A graphical representation of matrix $\Theta^{(j)}, j = 1, 2, 3, 4$ is given in the right panels.

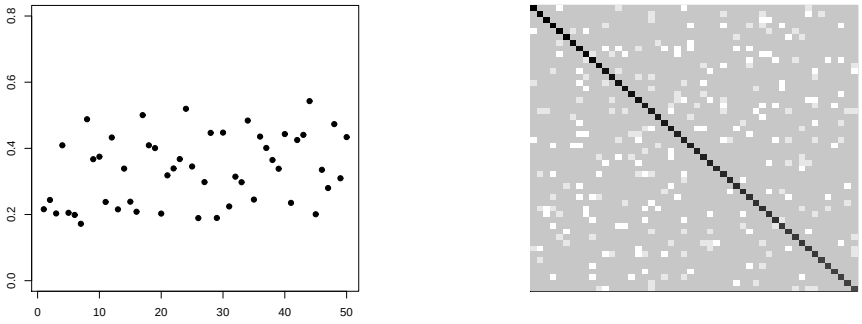

Figure 2: Marginal probabilities $\mathbb{P}(X^{(k)} = 1)$, for $k = 1, ..., 50$, estimated on a sample of size $n = 2500$ generated using the matrix $\Theta = \Theta^{(5)}$ are presented in the left panel. A graphical representation of matrix $\Theta^{(5)}$ is given in the right panel.

$$\mathcal{L}_{\lambda}^{\mathrm{G}_\nu} \quad = \quad \log|C_{\lambda}^{\nu}| - \mathrm{tr}(C_{\lambda}^{\nu} S_{\nu}) \text{ for } \nu = 1, 2, 3. \tag{18}$$

In (16), Θ_λ stands for the un-shrunk matrix derived under the sparsity structure inferred from method `BMNPseudo` with the sparsity parameter value λ, and $\tilde{x}_i^{(k)}$ and $\mathcal{X}^k$ are as in (7). In (17), $\Theta_\lambda^{\mathrm{Po}}$ is the matrix of coefficients obtained using a Poisson log-linear model under the constrained induced by this sparsity structure, and $P(\mathbf{x}, \Theta)$ is as in (1). Lastly, in (18), $S_1 = (\mathrm{Cov}(\mathbf{Z}) + \mathrm{diag}(1/3))$, $S_2 = \mathrm{Cov}(\mathbf{Z})$, $S_3 = \mathrm{Cor}(\mathbf{Z})$ and C_λ^ν is defined as

$$C_{\lambda}^{\nu} = \arg \max_{M \in \mathcal{M}_{\lambda}^{+}} \{ \log|M| - \mathrm{tr}(M S_{\nu}) \},$$

where $\mathcal{M}_{\lambda}^{+} = \{M \succ 0 : M_{k,\ell} = 0 \text{ for couples } (k, \ell) \text{ such that } (\Theta_\lambda)_{k,\ell} = 0\}$ (with Θ_λ as in (16)).

Figure 3 shows the corresponding deviances. It can be seen that using the Gaussian approximate log-likelihood based on the covariance matrix with the additional 1/3 term on the diagonal results in a deviance which is quite far from the exact one. Furthermore, the deviance obtained with the covariance matrix (without adding the 1/3 term on the diagonal) equals that obtained with the correlation matrix, and both are closer to the exact deviance. Finally, the deviance of the pseudo-(log)-likelihood is always greater than the exact deviance. Using half the pseudo-likelihood corrects this undesirable effect in most cases.

These results should obviously be considered with caution. Even if we tried to use various Θ matrices to generate the data (and the conclusions were consistently the same), a theoretical study would be needed to confirm these empirical findings.

3.4 Performance evaluation; the case $p = 10$

Let us first consider the performances achieved by the oracle models (Tables 1 and 2). Overall, methods `BMNPseudo` and `GaussCor` achieve good performances in terms of accuracy and F1-score. It is also noteworthy that the computational time is much higher for `BMNPseudo`, while

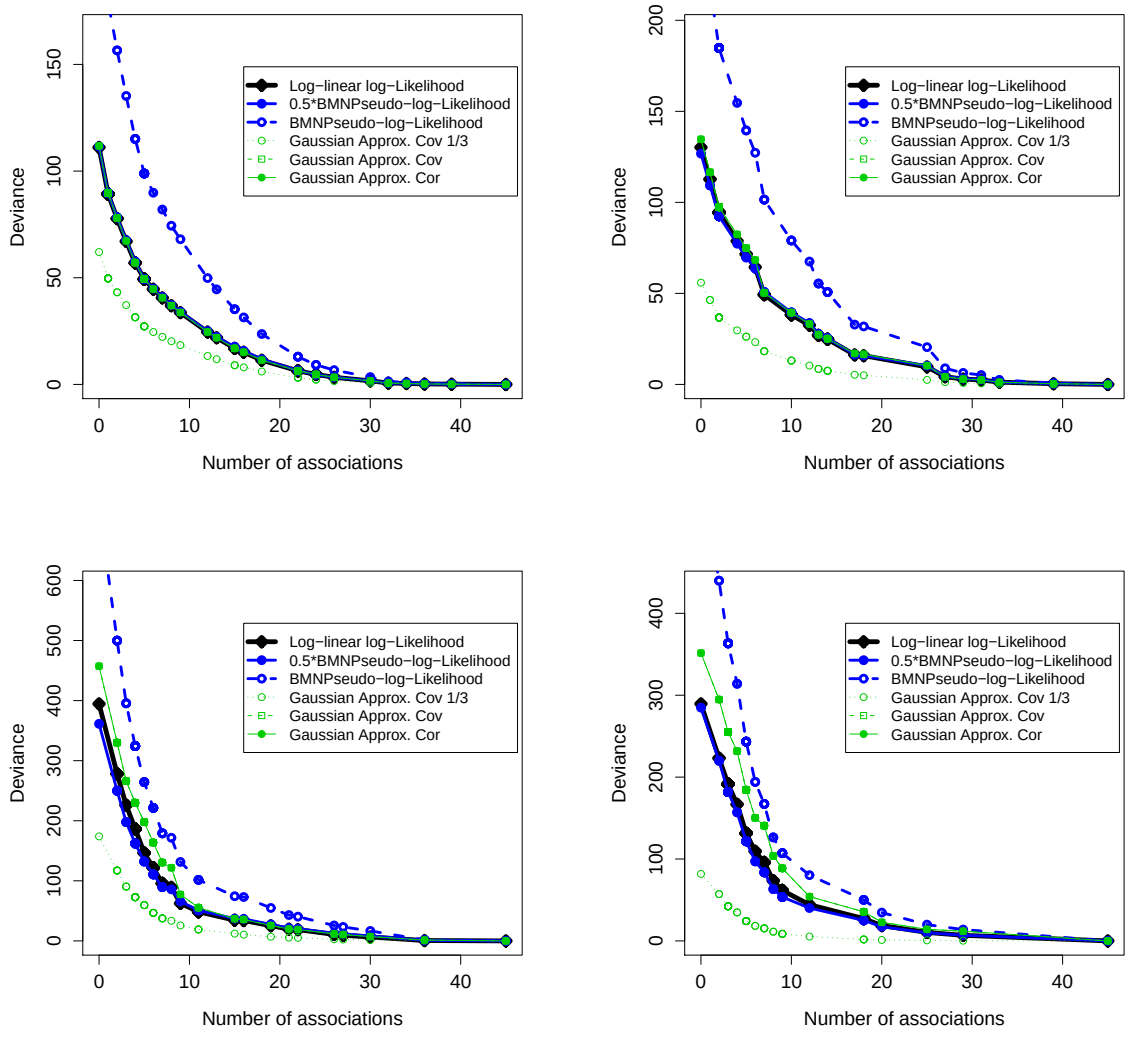

Figure 3: Approximates for the Ising deviance. Deviances were computed for each model selected by BMNPseudo for various values of the sparsity parameter, on samples of size $n = 500$ generated with matrices $\Theta^{(1)}$ (upper left corner), $\Theta^{(2)}$ (upper right corner), $\Theta^{(3)}$ (lower left corner), and $\Theta^{(4)}$ (lower right corner). Deviances were computed using the exact log-linear log-likelihood (solid black line, solid circles), the pseudo-likelihood (dashed blue line, circles), half the pseudo-likelihood (solid blue line, solid circles), and the Gaussian approximate log likelihoods based on the covariance matrix with an additional 1/3 term on the diagonal (dotted green line, circles), the covariance matrix (dashed green line, squares) and the correlation matrix (solid green line, solid circles)(see (16)-(18) for the corresponding formula).

overall performances of `GaussCor` appear to be slightly higher (especially under the fourth simulation design).

When focusing on the three `GaussApprox` methods, we observed the following ranking

$$\texttt{GaussCor} \geq \texttt{GaussCov} \geq \texttt{GaussCov 1/3}.$$

Consequently, and to save space, methods `GaussCov` and `GaussCov 1/3` will not be considered in the evaluation of the models selected via the BIC procedure. It is still interesting to note that `GaussCor`$\geq$`GaussCov` although we observed that the approximate deviances under these two methods were equal and close to the exact ones, which turn out to be an insufficient condition for achieving good performances.

Turning our attention to the evaluation of models selected with the BIC procedure (Tables 3 and 4), a first observation is that, as suggested by the results of Section 3.3, computing the BIC with half the pseudo-likelihood (rather than the pseudo-likelihood itself) results in better models in most cases. Moreover, from the comparisons of the results of Tables 1 and 2 and Tables 3 and 4, as n grows, the BIC procedure appears to enable to select models achieving performances similar to those achieved by the "oracle" models. Moreover, the computation of the un-shrunk estimates with method `BMNPseudo` appears to be very slow (the oracle models were much faster to compute than the models selected by the BIC approach for this particular method, especially when the sample size is small and in the fourth simulation design).

Overall, `SepLogit OR`, `SepLogit AND` and `GaussCor` are the best methods, closely followed by `BMNPseudo 1/2`. Lastly, among these candidate methods, `GaussCor` is the fastest.

We should lastly mention that method `SepLogit` was further tested using standardized covariates in each ℓ_1-penalized logistic regression models (results not shown). To motivate this choice, we may mention that this is the default option in package `glmnet` as this approach is often adopted in applications when using ℓ_1-penalization (see Koh et al. (2007) for instance); its suitability in our context of binary variables was yet questionable. Interestingly, this approach yielded results very similar to those obtained via the "standard" one on data generated using matrices $\Theta^{(1)}$, $\Theta^{(2)}$ and $\Theta^{(3)}$ and slightly better when using matrix $\Theta^{(4)}$ (which however corresponds to the situation where we observed the greatest variability in the performances of every method).

3.5 Performance evaluation; the case $p = 50$

To save space, we only present here the performances achieved by models selected via the BIC procedure, on samples of size $n = 500$ and $n = 2500$ generated using either matrix $\text{diag}(\Theta^{(j)}, \Theta^{(j)}, \Theta^{(j)}, \Theta^{(j)}, \Theta^{(j)})$, for $j = 1, 2, 3$ or matrix $\Theta^{(5)}$. Moreover, the results obtained in the case $p = 10$ especially show that method `BMNPseudo` can be quite slow, and that it does not outperform method `SepLogit`. Lastly, among the methods relying on a Gaussian approximate of the Ising likelihood, method `GaussCor` was observed to be the best. Therefore, in order to save computational time, only methods `GaussCor` and `SepLogit` were considered in the case $p = 50$.

Results are presented in Table 5. They are consistent with what was observed in the case $p = 10$. More precisely, methods `SepLogit` and `GaussCor` achieve comparable performances. Regarding computational time, `GaussCor` is still significantly faster than `SepLogit`.

Table 1: Evaluation of the "oracle models"; the case $p = 10$. Means (computed over 50 runs) are given for the computational time needed to compute the models on a grid of 50 equally-spaced λ values as well as the number of detected associations (POS), the false positive rate (FPR), the true positive rate (TPR, which equals the recall, REC), the precision (PRE), the accuracy (Acc.) and the F1 score corresponding to the "oracle" model.

Method	Time (s)	POS	FPR	TPR†	PRE	Acc.	F1 score
Data generated with $\Theta = \Theta^{(1)}$							
$n = 100$							
BMNPseudo	6.47	3.22	0.034	0.202	0.724	0.796	0.291
GaussCov 1/3	0.48	1.84	0.010	0.150	0.935	0.804	0.214
GaussCov	0.48	2.02	0.012	0.160	0.933	0.804	0.220
GaussCor	0.48	1.98	0.011	0.158	0.937	0.804	0.219
$n = 500$							
BMNPseudo	20.20	6.96	0.031	0.588	0.883	0.884	0.676
GaussCov 1/3	0.72	7.02	0.033	0.588	0.877	0.883	0.674
GaussCov	0.72	7.00	0.031	0.590	0.881	0.884	0.677
GaussCor	0.72	7.02	0.031	0.592	0.881	0.885	0.678
$n = 2500$							
BMNPseudo	82.65	9.48	0.003	0.938	0.991	0.984	0.961
GaussCov 1/3	0.96	9.36	0.002	0.928	0.992	0.982	0.957
GaussCov	0.96	9.40	0.002	0.932	0.992	0.983	0.959
GaussCor	0.96	9.40	0.002	0.932	0.992	0.983	0.959
Data generated with $\Theta = \Theta^{(2)}$							
$n = 100$							
BMNPseudo	20.17	3.62	0.024	0.278	0.864	0.821	0.376
GaussCov 1/3	0.54	3.72	0.025	0.284	0.867	0.821	0.376
GaussCov	0.54	3.92	0.028	0.294	0.857	0.821	0.386
GaussCor	0.55	3.60	0.025	0.274	0.870	0.820	0.366
$n = 500$							
BMNPseudo	23.16	7.02	0.014	0.654	0.950	0.912	0.755
GaussCov 1/3	0.79	7.44	0.02	0.674	0.927	0.912	0.761
GaussCov	0.79	7.46	0.019	0.678	0.93	0.913	0.765
GaussCor	0.86	8.24	0.017	0.766	0.938	0.935	0.832
$n = 2500$							
BMNPseudo	83.76	9.74	0.004	0.960	0.987	0.988	0.972
GaussCov 1/3	0.96	9.86	0.008	0.958	0.975	0.984	0.964
GaussCov	0.97	9.70	0.005	0.954	0.985	0.986	0.967
GaussCor	0.99	10.04	0.002	0.998	0.995	0.998	0.996

† TPR=REC.

Table 2: Evaluation of the "oracle models"; the case $p = 10$. Means (computed over 50 runs) are given for the computational time needed to compute the models on a grid of 50 equally-spaced λ values as well as the number of detected associations (POS), the false positive rate (FPR), the true positive rate (TPR, which equals the recall, REC), the precision (PRE), the accuracy (Acc.) and the F1 score corresponding to the "oracle" model.

Method	Time (s)	POS	FPR	TPR[†]	PRE	Acc.	F1 score
Data generated with $\Theta = \Theta^{(3)}$							
$n = 100$							
BMNPseudo	35.08	6.26	0.018	0.564	0.929	0.889	0.678
GaussCov 1/3	0.72	6.54	0.021	0.582	0.921	0.891	0.687
GaussCov	0.72	6.60	0.022	0.584	0.917	0.891	0.689
GaussCor	0.78	7.66	0.028	0.668	0.893	0.904	0.746
$n = 500$							
BMNPseudo	21.02	8.76	0.011	0.838	0.966	0.956	0.889
GaussCov 1/3	0.84	8.92	0.027	0.796	0.913	0.933	0.836
GaussCov	0.87	8.88	0.017	0.828	0.946	0.948	0.874
GaussCor	0.96	9.68	0.006	0.948	0.981	0.984	0.963
$n = 2500$							
BMNPseudo	80.60	10.02	0.002	0.994	0.993	0.997	0.993
GaussCov 1/3	0.92	10.84	0.035	0.962	0.894	0.964	0.923
GaussCov	0.97	10.10	0.007	0.984	0.976	0.991	0.979
GaussCor	0.99	10.00	0.001	0.998	0.998	0.999	0.998
Data generated with $\Theta = \Theta^{(4)}$							
$n = 100$							
BMNPseudo	39.89	5.46	0.055	0.212	0.816	0.635	0.312
GaussCov 1/3	0.60	6.84	0.084	0.245	0.775	0.633	0.333
GaussCov	0.60	5.16	0.053	0.199	0.800	0.631	0.297
GaussCor	0.60	6.86	0.079	0.253	0.761	0.639	0.357
$n = 500$							
BMNPseudo	183.22	12.94	0.084	0.566	0.859	0.768	0.668
GaussCov 1/3	0.68	12.36	0.08	0.541	0.853	0.760	0.651
GaussCov	0.69	13.02	0.089	0.563	0.840	0.764	0.665
GaussCor	0.75	13.64	0.055	0.643	0.905	0.818	0.745
$n = 2500$							
BMNPseudo	255.87	15.06	0.058	0.714	0.906	0.846	0.796
GaussCov 1/3	0.76	14.68	0.072	0.674	0.880	0.820	0.759
GaussCov	0.78	15.16	0.072	0.700	0.884	0.832	0.778
GaussCor	0.85	15.70	0.037	0.776	0.944	0.884	0.848

[†] TPR=REC.

Table 3: Evaluation of the models selected by the BIC procedure; the case $p = 10$. Means (computed over 50 runs) are given for the computational time needed to compute the models on a grid of 50 equally-spaced λ values as well as for the number of detected associations (POS), the false positive rate (FPR), the true positive rate (TPR, which equals the recall, REC), the precision (PRE), the accuracy (Acc.) and the F1 score corresponding to the model selected by the BIC procedure.

Method	Time (s)	POS	FPR	TPR[†]	PRE	Acc.	F1 score
Data generated with $\Theta = \Theta^{(1)}$							
$n = 100$							
SepLogit OR	21.94	2.86	0.043	0.136	0.540	0.775	0.234
SepLogit AND	21.94	2.12	0.029	0.112	0.585	0.780	0.216
BMNPseudo	14.20	8.16	0.141	0.322	0.424	0.740	0.355
BMNPseudo 1/2	14.20	2.60	0.039	0.122	0.492	0.774	0.233
GaussCor	1.12	2.56	0.035	0.132	0.572	0.780	0.235
$n = 500$							
SepLogit OR	22.91	4.80	0.018	0.416	0.885	0.856	0.549
SepLogit AND	22.91	4.12	0.014	0.364	0.892	0.848	0.504
BMNPseudo	42.43	9.48	0.079	0.670	0.725	0.865	0.687
BMNPseudo 1/2	42.43	4.60	0.015	0.408	0.895	0.857	0.550
GaussCor	1.16	4.48	0.014	0.400	0.895	0.856	0.539
$n = 2500$							
SepLogit OR	27.89	9.52	0.006	0.93	0.979	0.980	0.952
SepLogit AND	27.89	9.32	0.004	0.918	0.986	0.979	0.949
BMNPseudo	176.09	11.66	0.055	0.972	0.846	0.951	0.901
BMNPseudo 1/2	176.09	9.36	0.005	0.920	0.985	0.979	0.948
GaussCor	1.19	9.30	0.005	0.912	0.983	0.976	0.943
Data generated with $\Theta = \Theta^{(2)}$							
$n = 100$							
SepLogit OR	23.45	4.68	0.058	0.266	0.618	0.792	0.361
SepLogit AND	23.45	2.76	0.027	0.182	0.687	0.797	0.302
BMNPseudo	215.65	9.82	0.154	0.444	0.482	0.757	0.443
BMNPseudo 1/2	215.65	3.22	0.033	0.208	0.688	0.799	0.325
GaussCor	1.11	3.80	0.041	0.236	0.686	0.798	0.341
$n = 500$							
SepLogit OR	33.18	7.20	0.018	0.658	0.922	0.910	0.758
SepLogit AND	33.18	6.18	0.007	0.592	0.960	0.904	0.720
BMNPseudo	97.70	10.82	0.087	0.776	0.735	0.882	0.747
BMNPseudo 1/2	97.70	6.56	0.014	0.606	0.934	0.901	0.720
GaussCor	1.16	7.02	0.014	0.652	0.931	0.912	0.758
$n = 2500$							
SepLogit OR	34.52	10.28	0.011	0.990	0.966	0.989	0.977
SepLogit AND	34.52	9.92	0.005	0.976	0.985	0.991	0.980
BMNPseudo	175.80	11.70	0.049	0.998	0.872	0.961	0.926
BMNPseudo 1/2	175.80	10.16	0.013	0.972	0.960	0.984	0.965
GaussCor	1.30	10.06	0.005	0.988	0.984	0.993	0.985

[†] TPR=REC.

Table 4: Evaluation of the models selected by the BIC procedure; the case $p = 10$. Means (computed over 50 runs) are given for the computational time needed to compute the models on a grid of 50 equally-spaced λ values as well as the number of detected associations (POS), the false positive rate (FPR), the true positive rate (TPR, which equals the recall, REC), the precision (PRE), the accuracy (Acc.) and the F1 score corresponding to the model selected by the BIC procedure.

Method	Time (s)	POS	FPR	TPR[†]	PRE	Acc.	F1 score
Data generated with $\Theta = \Theta^{(3)}$							
$n = 100$							
SepLogit OR	30.86	7.86	0.055	0.594	0.775	0.867	0.660
SepLogit AND	30.86	5.44	0.021	0.470	0.868	0.866	0.599
BMNPseudo	276.48	12.18	0.147	0.702	0.613	0.819	0.640
BMNPseudo 1/2	276.48	6.44	0.031	0.534	0.856	0.872	0.638
GaussCor	1.26	7.42	0.037	0.612	0.834	0.885	0.697
$n = 500$							
SepLogit OR	33.10	10.14	0.027	0.918	0.912	0.960	0.912
SepLogit AND	33.10	8.90	0.009	0.858	0.967	0.961	0.906
BMNPseudo	163.20	12.78	0.098	0.936	0.760	0.910	0.830
BMNPseudo 1/2	163.20	9.80	0.030	0.876	0.906	0.949	0.885
GaussCor	1.26	9.78	0.014	0.930	0.957	0.974	0.940
$n = 2500$							
SepLogit OR	40.76	10.30	0.009	1.000	0.973	0.993	0.986
SepLogit AND	40.76	10.02	0.001	1.000	0.998	0.999	0.999
BMNPseudo	176.22	11.14	0.033	1.000	0.914	0.975	0.951
BMNPseudo 1/2	176.22	10.18	0.005	1.000	0.984	0.996	0.992
GaussCor	1.26	10.12	0.003	1.000	0.989	0.997	0.994
Data generated with $\Theta = \Theta^{(4)}$							
$n = 100$							
SepLogit OR	30.63	20.96	0.443	0.497	0.452	0.532	0.470
SepLogit AND	30.63	14.88	0.322	0.343	0.440	0.537	0.383
BMNPseudo	193.43	27.16	0.577	0.640	0.446	0.515	0.523
BMNPseudo 1/2	193.43	18.70	0.405	0.431	0.440	0.526	0.431
GaussCor	1.87	20.72	0.452	0.473	0.432	0.516	0.445
$n = 500$							
SepLogit OR	36.10	10.48	0.038	0.500	0.916	0.767	0.642
SepLogit AND	36.10	8.30	0.012	0.421	0.967	0.749	0.583
BMNPseudo	1090.87	13.52	0.107	0.565	0.815	0.755	0.657
BMNPseudo 1/2	1090.87	9.84	0.044	0.458	0.897	0.746	0.600
GaussCor	1.80	10.70	0.025	0.529	0.945	0.787	0.675
$n = 2500$							
SepLogit OR	52.32	15.26	0.048	0.737	0.922	0.861	0.817
SepLogit AND	52.32	13.44	0.016	0.685	0.971	0.858	0.802
BMNPseudo	2299.82	17.40	0.122	0.748	0.829	0.823	0.783
BMNPseudo 1/2	2299.82	14.72	0.056	0.698	0.905	0.840	0.786
GaussCor	1.86	14.76	0.026	0.741	0.957	0.876	0.834

[†] TPR=REC.

Table 5: Evaluation of the models selected by the BIC procedure; the case $p = 50$. Means (computed over 50 runs) are given for the computational time needed to compute the models on a grid of 50 equally-spaced λ values as well as the number of detected associations (POS), the false positive rate (FPR), the true positive rate (TPR, which equals the recall, REC), the precision (PRE), the accuracy (Acc.) and the F1 score corresponding to the model selected by the BIC procedure.

Method	Time (s)	POS	FPR	TPR[†]	PRE	Acc.	F1 score
Data generated with $\Theta = \mathbf{diag}(\Theta^{(1)}, \Theta^{(1)}, \Theta^{(1)}, \Theta^{(1)}, \Theta^{(1)})$							
$n = 500$							
SepLogit OR	236.21	39.00	0.016	0.399	0.515	0.960	0.447
SepLogit AND	236.21	30.10	0.011	0.352	0.588	0.963	0.438
GaussCor	31.85	34.06	0.013	0.376	0.562	0.962	0.446
$n = 2500$							
SepLogit OR	248.74	54.48	0.006	0.938	0.864	0.991	0.898
SepLogit AND	248.74	50.64	0.004	0.921	0.910	0.993	0.915
GaussCor	28.86	52.34	0.005	0.918	0.881	0.991	0.898
Data generated with $\Theta = \mathbf{diag}(\Theta^{(2)}, \Theta^{(2)}, \Theta^{(2)}, \Theta^{(2)}, \Theta^{(2)})$							
$n = 500$							
SepLogit OR	285.91	53.36	0.017	0.661	0.626	0.970	0.640
SepLogit AND	285.91	34.14	0.007	0.527	0.778	0.974	0.626
GaussCor	32.75	45.42	0.011	0.642	0.717	0.975	0.673
$n = 2500$							
SepLogit OR	304.09	59.26	0.008	0.988	0.837	0.991	0.905
SepLogit AND	304.09	50.80	0.003	0.950	0.937	0.995	0.943
GaussCor	29.00	54.40	0.004	0.995	0.917	0.996	0.954
Data generated with $\Theta = \mathbf{diag}(\Theta^{(3)}, \Theta^{(3)}, \Theta^{(3)}, \Theta^{(3)}, \Theta^{(3)})$							
$n = 500$							
SepLogit OR	305.44	68.66	0.019	0.936	0.685	0.980	0.790
SepLogit AND	305.44	47.04	0.005	0.834	0.888	0.989	0.859
GaussCor	30.34	60.34	0.010	0.968	0.811	0.989	0.880
$n = 2500$							
SepLogit OR	380.86	57.72	0.007	1.000	0.868	0.994	0.929
SepLogit AND	380.86	51.50	0.001	1.000	0.971	0.999	0.985
GaussCor	25.48	50.48	0.000	1.000	0.991	1.000	0.995
Data generated with $\Theta = \Theta^{(5)}$							
$n = 500$							
SepLogit OR	271.84	95.20	0.017	0.610	0.802	0.945	0.692
SepLogit AND	271.84	71.00	0.007	0.508	0.894	0.944	0.647
GaussCor	74.71	86.96	0.013	0.582	0.839	0.946	0.685
$n = 2500$							
SepLogit OR	307.38	129.00	0.008	0.960	0.931	0.989	0.945
SepLogit AND	307.38	117.48	0.002	0.922	0.981	0.990	0.950
GaussCor	65.41	123.68	0.007	0.927	0.938	0.986	0.932

[†] TPR=REC.

3.6 Agreement between the compared methods

One way to measure the agreement between two selected models is to compare them with their intersection. Let $\mathcal{G}_1 = (V, E_1)$ and $\mathcal{G}_2 = (V, E_2)$ be the two graphs to be compared. Further let $E^\star = E_1 \cap E_2$ and denote by $\sharp E$ the cardinality of a set E of edges. To compare graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we will consider the quantities

$$\kappa(\mathcal{G}_1, \mathcal{G}_2) = \frac{\sharp E^\star}{\min(\sharp E_1, \sharp E_2)}, \tag{19}$$

$$\bar{\kappa}(\mathcal{G}_1, \mathcal{G}_2) = (\sharp E_1 - \sharp E^\star) + (\sharp E_2 - \sharp E^\star), \tag{20}$$

as measures of agreement and disagreement respectively ($\bar{\kappa}$ is the cardinality of the symmetric difference between E_1 and E_2). Observe that according to these measures, the saturated graph would both agree and disagree "a lot" with any sub-graph.

Results are presented in Table 6 for $p = 10$ and $\Theta = \{\Theta^{(3)}, \Theta^{(4)}\}$ which correspond to the two extreme situations in terms of signal-to-noise ratio. Overall, `BMNPseudo` is closer to `SepLogit` than `GaussCor`, what was expected given the respective principles of the methods. Moreover, agreement [resp. disagreement] between the various models is higher [resp. lower] when the signal-to-noise ratio is high, that is when $n = 2500$ and/or $\Theta = \Theta^{(3)}$. When the signal-to-noise ratio is low and models are quite different, a natural question is how to get the best model. Intersecting two models is one of the candidate approaches. For the sake of brevity, we do not present the complete results here, but intersecting `GaussCor` and `SepLogit OR` for instance resulted in quite conservative models that generally achieved better performances than either `GaussCor` or `SepLogit OR` (in terms of accuracy and F1-score).

Models derived under method `SepLogit` with standardized covariates were also compared to the other models (results not shown). Overall, we observed very good agreement between the standard approach and the standardized approach (especially in the case of high signal-to-noise ratio). We also observed slightly better agreements between these models and those derived under method `GaussCor`, especially on datasets generated using matrix $\Theta^{(4)}$.

3.7 Comparison of the conditional odds-ratios estimates

Both methods `SepLogit` and `BMNPseudo` have been empirically shown to yield appropriate estimates for the conditional odds-ratios. On the other hand, it is rather unclear whether estimates derived from methods relying on the Gaussian approximation would be consistent. We therefore conclude this simulation study with a simple study to check it.

To avoid interpretation difficulties related to the model selection accuracy, coefficient estimates were computed under the sparsity structure of the true models and compared with the true coefficients (this corresponds to the situation where the true sparsity structure is known). To do so, we used an approach similar to the one used to get un-shrunk estimates for the BIC procedure.

The mean squared errors for the conditional log-odds ratios, which we defined here as

$$\text{MSE} = 1000 \times \frac{\sum_{k>\ell}(\hat{\theta}_{k,\ell} - \theta_{k,\ell})^2}{\sum_{k>\ell} \mathbb{1}\{\theta_{k,\ell} \neq 0\}}, \tag{21}$$

Table 6: Results of the simulation study: agreement evaluation. Means (along with standard deviations) of the criteria defined in (19)-(20) were obtained from 50 runs for various sample sizes n and matrices Θ.

Comparisons	Agreement κ		Disagreement $\bar{\kappa}$	
Data generated with $\Theta = \Theta^{(3)}$, $n = 100$				
SepLogit OR SepLogit AND	1.000	(0.000)	2.420	(1.605)
SepLogit OR BMNPseudo	0.970	(0.071)	2.140	(1.807)
SepLogit AND BMNPseudo	0.965	(0.099)	1.720	(1.578)
BMNPseudo GaussCor	0.941	(0.087)	2.940	(1.463)
SepLogit OR GaussCor	0.941	(0.077)	2.480	(1.632)
SepLogit AND GaussCor	0.940	(0.086)	3.060	(1.476)
Data generated with $\Theta = \Theta^{(3)}$, n=2500				
SepLogit OR SepLogit AND	1.000	(0.000)	0.280	(0.497)
SepLogit OR BMNPseudo	1.000	(0.000)	0.160	(0.370)
SepLogit AND BMNPseudo	0.998	(0.013)	0.200	(0.495)
BMNPseudo GaussCor	1.000	(0.000)	0.100	(0.303)
SepLogit OR GaussCor	1.000	(0.000)	0.220	(0.418)
SepLogit AND GaussCor	0.998	(0.013)	0.140	(0.405)
Data generated with $\Theta = \Theta^{(4)}$, $n = 100$				
SepLogit OR SepLogit AND	1.000	(0.000)	6.080	(2.146)
SepLogit OR BMNPseudo	0.952	(0.048)	4.800	(2.711)
SepLogit AND BMNPseudo	0.957	(0.053)	5.400	(2.148)
BMNPseudo GaussCor	0.967	(0.054)	6.740	(2.863)
SepLogit OR GaussCor	0.896	(0.081)	8.380	(2.900)
SepLogit AND GaussCor	0.931	(0.081)	9.900	(3.112)
Data generated with $\Theta = \Theta^{(4)}$, n=2500				
SepLogit OR SepLogit AND	1.000	(0.000)	1.820	(1.424)
SepLogit OR BMNPseudo	0.981	(0.031)	1.880	(1.335)
SepLogit AND BMNPseudo	0.983	(0.030)	1.900	(1.359)
BMNPseudo GaussCor	0.950	(0.050)	2.900	(1.474)
SepLogit OR GaussCor	0.972	(0.035)	2.220	(1.183)
SepLogit AND GaussCor	0.980	(0.035)	2.200	(1.666)

Table 7: Results of the simulation study: evaluation of the conditional log-odds-ratios estimation. Means of the mean squared error (MSE) defined in (21) were obtained from 50 runs for various sample sizes n and matrices Θ. Three methods were evaluated: the genuine method of Banerjee et al. (2008) GaussCov 1/3, its modification relying on the correlation matrix GaussCor and the method relying on multiple logistic regressions SepLogit.

Sample size	GaussCov 1/3	GaussCor	SepLogit
Data generated with $\Theta = \Theta^{(1)}$			
$n = 500$	2.613	2.073	1.957
$n = 2500$	2.140	0.451	0.435
Data generated with $\Theta = \Theta^{(2)}$			
$n = 500$	14.568	7.743	6.175
$n = 2500$	14.342	6.573	1.201
Data generated with $\Theta = \Theta^{(3)}$			
$n = 500$	58.502	29.197	10.704
$n = 2500$	57.451	26.779	1.739
Data generated with $\Theta = \Theta^{(4)}$			
$n = 500$	98.304	68.034	50.007
$n = 2500$	97.402	65.973	12.794

are reported in Table 7 for methods `SepLogit`, `GaussCov 1/3` and `GaussCor` in the case $p = 10$ and for samples of sizes $n = 500$ and $n = 2500$ (for `SepLogit` each coefficient $\hat{\theta}_{k,\ell}$ was set as the mean of the two coefficients returned by the two constrained logistic regressions involved in this method). It can be observed that overall neither `GaussCov 1/3` nor `GaussCor` led to consistent estimates for the $\theta_{k,\ell}$ coefficients. These methods (especially `GaussCor`) should therefore be combined with other methods (for instance `SepLogit`) when estimates of the conditional odds-ratios are needed.

Inconsistency of the estimates derived under method `GaussCor` can be explained as follows. Since this method relies on the correlation matrix, it is closely related to the method consisting in performing linear regressions at each node (as shown by Meinshausen and Bühlmann (2006) in the Gaussian case). Therefore, the coefficients returned by this method are related to the coefficients $\gamma_{k,\ell}$ involved in the linear model

$$\mathbb{E}(X^{(k)}|\mathbf{x}^{(-k)}) = \mathbb{P}(X^{(k)} = 1|\mathbf{x}^{(-k)}) = \gamma_k + \sum_{\ell \neq k} \gamma_{k,\ell} x^{(\ell)}. \tag{22}$$

Clearly, coefficients $\gamma_{k,\ell}$ are quite different from the conditional log odds-ratios $\theta_{k,\ell}$ involved in the Ising model (see (1)).

4 Application to the search for associations among causes of death

The general objective of the application in this example is to detect associations between causes of death and identify possibly relevant groupings of causes contributing to the death.

4.1 Description of the data

The dataset we used consists in causes of death recorded in all death certificates ($n = 535\,684$) in France for the year 2005. In France, death certification (compulsory with 100% coverage) conforms to the WHO guideline: the death process is described starting from the underlying causes of death and ending with the immediate cause of death; other contributing causes of death are also recorded. All these causes were considered in the analysis. Causes are further coded according to the International Classification of Diseases (ICD)(in 2005, the tenth revision (World Heath Organization, 1994)). The total number of code categories is large (about 4000 codes used) but for this analysis we applied a simplified classification of 59 entities according to the Eurostat shortlist (EUROSTAT, 2002) (see Appendix A for the classification used). Therefore, $p = 59$ and each death certificate can be regarded as a vector $\mathbf{x} = (x^{(1)}, ..., x^{(59)})$ in $\{0, 1\}^{59}$, where, for all $k = 1, ..., 59$, $x^{(k)} = 1$ if and only if the k-th class is recorded on the death certificate. About 11% of the certificates had more than five causes of death, 14% had four causes, 26% had three, 30% had two and 18% had a sole cause of death. The frequencies of each cause are reported in Appendix A; the most frequent causes of death were, in decreasing order, heart failure, ischaemic heart diseases, cerebrovascular diseases, hypertensive diseases, pneumonia, diabetes, lung cancer and senility.

4.2 Graphical model analysis

Most common causes of death clearly depend upon age and gender, and a natural question is whether associations among causes of death also vary with age and gender. We then decided to split the whole population into strata defined by gender and age The complete analysis of every stratum is out of the scope of the present paper, and we only present here the results from the analysis of two sub-groups, namely those of males aged between 15 and 24 (2918 observations) and males aged between 45 and 64 (57045 observations).

First considering the group of age 15-24, we applied `GaussCor`, `BMNPseudo`, `SepLogit AND`, and `SepLogit OR` after selecting the sparsity parameter according to the BIC procedure described above. This yielded models with 113, 107, 61 and 129 associations respectively. Good agreement was observed between models derived under methods `SepLogit` and `BMNPseudo` (we had $\kappa = 0.935$ for the comparison between `BMN` and `SepLogit OR` for instance). However, the model derived under method `GaussCor` was slightly different from the three other models (we had $\kappa = 0.700$ for the comparison between `GaussCor` and `SepLogit OR` and $\kappa = 0.918$ for the comparison between `GaussCor` and `SepLogit AND`). More precisely, the model obtained with method `GaussCor` entailed many more positive associations, a few of which corresponding to variables co-occurring only once in the sub-group. This suggests that method `GaussCor` might be a little too sensitive to positive associations, especially when the signal-to-noise ratio is low (in this particular study, the signal-to-noise ratio is low due to highly unbalanced variables). Regarding computational time, 1.6 second was needed to compute method `GaussCor` while it took 19628 and 876 seconds for computing methods `BMNPseudo` and `SepLogit` respectively (analyses were performed on the Windows machine). For these latter two methods, we were not able to conduct the analysis with the choice $\lambda^{\min} = \lambda^{\max}/1000$, and we had to select $\lambda^{\min} = \lambda^{\max}/50$ and $\lambda^{\min} = \lambda^{\max}/100$ for methods `BMNPseudo` and `SepLogit` respectively. It is also noteworthy that the computational time needed for methods `BMNPseudo` and `SepLogit` is mostly due to the computation of the un-shrunk estimates (necessary for the

derivation of the BIC); omitting this step, the computational time using method `BMNPseudo` [resp. `SepLogit`] is reduced to 4598 seconds [resp. 198 seconds].

Figure 4 shows the final retained model for the group 15-24, which is the intersection of the models derived under methods `SepLogit OR` and `GaussCor`. Apart from the obvious association between depression and suicide, the strongest (positive) associations identified were between diabetes and other endocrinal diseases, colorectal cancer and metastasis, septicemia and pneumonia, and between diseases of arteries, arterioles and capillaries and cerebrovascular diseases. The strong negative associations between transport accidents and all other conditions, and between suicide and most other conditions (except depression and other mental disorders) is also worth noting. These latter associations correspond to well-known sequences of causes leading to death and most of those present in the figure have strong biological plausibility.

In the analysis of the older group, we only applied methods `SepLogit` and `GaussCor` (to save computational time), which took 15241 and 4.8 seconds respectively. In this case, we had to use $\lambda^{\min} = \lambda^{\max}/50$ for method `SepLogit`. Moreover, when omitting the computation of the un-shrunk estimates, the computational time using method `SepLogit` reduced to 1943 seconds. 600, 778 and 708 associations were detected by method `SepLogit AND`, `SepLogit OR` and `GaussCor` respectively. Good agreement was observed between the models (we had $\kappa = 0.933$ for the comparison between `GaussCor` and `SepLogit OR` and $\kappa = 0.953$ for the comparison between `GaussCor` and `SepLogit AND`), which tends to confirm that agreement between the models returned by methods `SepLogit` and `GaussCor` increases with the signal-to-noise ratio.

5 Discussion

In this paper we empirically compared several approximate methods designed to search for associations among binary variables. We observed that methods `SepLogit` and `BMNPseudo` achieved similar performances in terms of accuracy and F1-score, with a slight advantage to method `SepLogit`. Moreover, the models selected by both methods are very similar in most cases, as could be expected given the similarity between them. In terms of computational time, `SepLogit` appeared to be overall faster than `BMNPseudo`, but the two methods share the disadvantage of being quite slow to compute, especially for low values of the sparsity parameter.

For the method `BMNPseudo`, we observed that using half the pseudo-likelihood rather than the pseudo-likelihood itself when computing the BIC enables us to select better models in most cases. The multiplicative coefficient 1/2 might not be optimal in all situation and some adaptive coefficient might be derived from a theoretical study of the pseudo-likelihood. Alternatively, cross-validation could be considered at a cost of an increased computational time, which seems undesirable given the aforementioned lack of speed of this method. Moreover, the suitability of cross-validation for model selection remains questionable (Gao et al., 2009).

In terms of accuracy, the method proposed by Banerjee et al. (2008) was shown to be generally too conservative. We then proposed a slight modification, referred to as `GaussCor`, in which we remove the additive 1/3 term on the diagonal, and use the sample correlation matrix as a starting point for the algorithm. With these modifications, `GaussCor` combines good overall performances (comparable to the performances achieved by `SepLogit` and `BMNPseudo`)

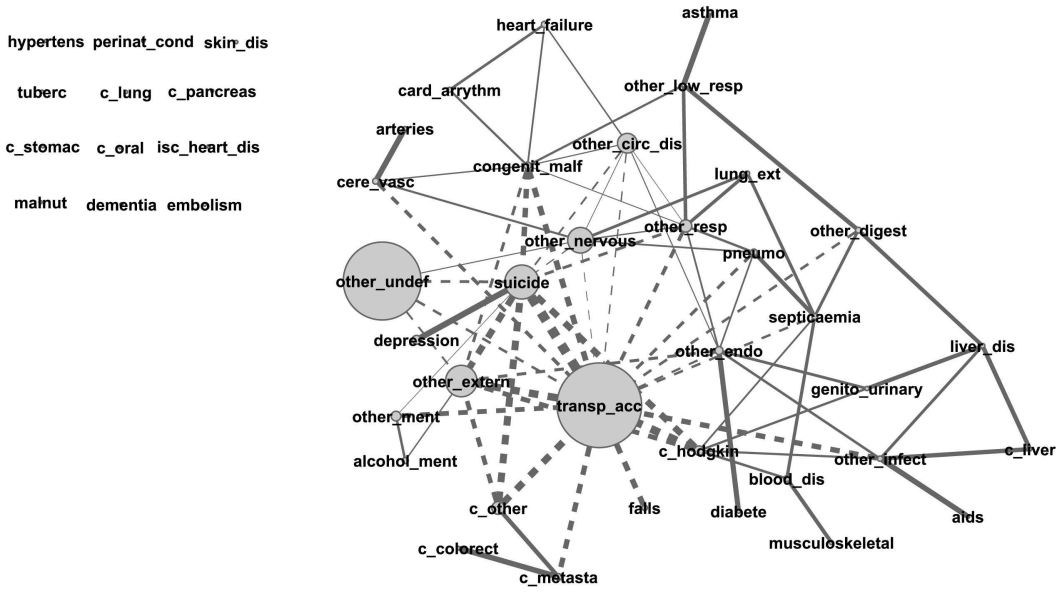

Figure 4: Graphical model obtained with Cytoscape on the real data set (males aged between 15 and 24 years). Positive associations (solid lines) and negative associations (dashed lines) are presented. The line widths of edges are proportional to the conditional log-odds-ratios (estimated using multiple logistic regressions built under the constraint implied by the retained model). For instance, the (absolute value of the) conditional log-odds-ratio was about 4.5 for the association depression/suicide, 3 for the association liver disease/other diseases of the digestive system, 1.7 for the association other infectious disease/other endocrinal disease, and 0.35 for the association other mental disorderd/suicide. Similarly, vertices are represented by balls with diameter related to the observed frequency of the causes of death. Transport accidents were reported on about 40% of the death certificates while falls were only reported on 1.2% of the death certificates. Conditions listed on the left side are not associated with any other condition or disease.

and exceptional computational speed. In particular, `GaussCor` was observed to be between 3 and 200 times faster than the other methods on simulated data. This speed is particularly desirable for handling truly high-dimensional datasets since the concurrent methods (`SepLogit` or `BMNPseudo`) might be dramatically slow in such cases. To be complete, we should mention that method `SepLogit` could be implemented using other sparse logistic regression algorithms that might be faster than the `glmnet` R package (see Koh et al. (2007); Genkin et al. (2007); Lee et al. (2006) for instance). However, we think that the comparison conducted here was fair since both R packages `glmnet` and `glasso` rely on a coordinate descent algorithm (Friedman et al., 2008b).

Interestingly, we also observed that the models selected by methods `GaussCor` and `SepLogit` can be significantly different, especially in the situation of low signal-to-noise ratio. On our real example, we decided to retain the intersection of the two selected models as the final model, which led to conservative but competitive models on simulated examples. However, other approaches can be considered and it would be interesting to further study how these methods can be optimally combined.

Approximate methods that use either multiple logistic regressions or the pseudo-likelihood have been shown to attain performances similar to those reached using exact inference at a lower computational cost (Höfling and Tibshirani, 2009). Our results suggest here that using Gaussian approximates of the Ising likelihood can ensure similar statistical performance at a greatly improved speed. In the absence of a theoretical justification for the good performances achieved by this method however, we can only claim here that `GaussCor` is a candidate method that can be recommended in some cases; a theoretical study might enable to have a better idea of what these cases are.

Appendix: The retained classification of causes of death

Disease/Condition	Label	ICD-10 codes	Count
Septicaemia	septicaemia	A40-A41	25713
tuberculosis	tuberc	A15-A19, B90	1720
aids and HIV infection	aids	B20-B24	1094
Other infectious disease	other_infect	A00-A14, A20-A39, A42-B19, B25-B89, B91-B99	14188
Oral cancer	c_oral	C00-C14	5076
Oesophageal cancer	c_oesoph	C15	4654
Stomach cancer	c_estomac	C16	5642
Colorectal cancer	c_colorect	C18-C21	19587
Liver cancer	c_liver	C22	8528
Pancreas cancer	c_pancreas	C25	8615
Larynx cancer	c_larynx	C32	2062
Lung cancer	c_lung	C33-C34	30642
Breast cancer	c_breast	C50	14439
Uterus cancer	c_uterus	C53-C55	3708
Prostate cancer	c_prostate	C61	13361
Bladder cancer	c_bladder	C67	5946
Hodgkin's disease and leukemia	c_hodgkin	C81-C96	15574
Secondary malignant neoplasm	c_metasta	C77-C79	50314
Other cancers	c_other	C17, C23-C24, C26-C31, C35-C49, C51-C52, C56-C60, C62-C66, C68-C76, C80, C97-D49	72943
Diseases of the blood	blood_dis	D50-D89	12955
Diabetes	diabetes	E10-E14	32704
Malnutrition	malnut	E40-E46	12713
Other endocrinal disease	other_endo	E00-E09, E15-E39, E47-E90	27490
Dementia	dementia	F01-F03	22966
Mental disorders due to use of alcohol	alcohol_ment	F10	12634
Mood disorders	depression	F30-F39	8628
Other mental disorders	other_ment	F00, F04-F09, F11-F29, F40-F99	15629
Parkinson's disease	parkinson	G20	8598
Alzheimer's disease	Alzheimer	G30	22568
Other diseases of the nervous system,the eye and adnexa	other_nervous	G00-G19, G21-G29, G31-H95	27028
Hypertensive diseases	hypertens	I10-I15	44117
Ischaemic heart diseases	isc_heart_dis	I20-I25	62071
Pulmonary embolism,phlebitis and thrombophlebitis	embolism	I26, I80-I82	16697
Cardiac arrhythmias	card_arrhytm	I47-I49	38020
Heart failure	heart_fail	I50	73268
Cerebrovascular diseases	cere_vasc	I60-I69	58161
Diseases of arteries, arterioles and capillaries	arteries	I70-I79	25567
Other diseases of the circulatory system	other_circ	I00-I09, I16-I19, I27-I46, I51-I59, I83-I99	80060
Influenza (other than avian influenza)	influenza	J10-J11	1192

Disease/Condition	Label	ICD-10 codes	Count
Pneumonia	pneumo	J12-J18	38677
Asthma and status asthmaticus	asthma	J45-J46	2886
Other chronic lower respiratory diseases	other_low_resp	J40-J44, J47	17739
Lung diseases due to external agents	lung_extern	J60-J70	10137
Other diseases of the respiratory system	other_resp	J00-J09, J19-J39, J48-J59, J71-J99	59123
Peptic ulcer	pept_ulcer	K25-K28	1965
Diseases of liver	liver_dis	K70-K77	20661
Other diseases of the digestive system	other_digest	K00-K24, K29-K69, K78-K99	34507
Diseases of the skin and subcutaneous tissue	skin_dis	L00-L99	11056
Diseases of the musculoskeletal system and connective tissue	musculoskeletal	M00-M99	9644
Diseases of the genitourinary system	genito_urinary	N00-N99	37683
Pregnancy, childbirth and the puerperium	pregnancy	O00-O99	67
Certain conditions originating in the perinatal period	perinat_cond	P00-P96	2100
Congenital malformations, deformations and chromosomal abnormalities	congenit_malf	Q00-Q99	2411
Senility	senility	R54	23646
Other symptoms and abnormal clinical findings, not elsewhere classified	other_undef	R00-R53, R55-R59	252979
Transport accidents	transp_acc	V01-V99	5686
Falls	falls	W00-W19	6217
Intentional self-harm	suicide	X60-X84	10900
Other external causes of morbidity and mortality	other_extern	W20-X59, X85-Y89	21813

References

A. Agresti. *Categorical Data Analysis.* John Wiley and Sons, 1990

O. Banerjee, L. El Ghaoui and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research* **9,** 485–516, 2008.

J. Besag. Statistical analysis of non-lattice data. *The Statistician* **24,** 179–195, 1975.

C. Dahinden, G. Parmigiani, M.C. Emerick and P. Bühlmann. Penalized likelihood for sparse contingency tables with an application to full-length cDNA libraries. *BMC Bioinformatics* **8,** 476, 2007.

J. Dahl, L. Vandenberghe and V. Roychowdhury. Covariance selection for non-chordal graphs via chordal embedding. *Optimization Methods and Software* **23,** 501–520, 2008.

A.P. Dempster. Covariance selection. *Biometrika* **32,** 95–108, 1972.

D.M. Edwards. *Introduction to Graphical Modelling.* New York: Springer, 2000

B. Efron, T. Hastie, I. Johnstone and R. Tibshirani. Least Angle Regression. *Annals of Statistics* **32,** 407–499, 2004.

EUROSTAT. *Health statistics. Atlas on mortality in the European Union (data 1994-96).* Luxembourg: European Communities, 2002

J. Friedman, T. Hastie and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9,** 432–441, 2008.

J. Friedman, T. Hastie and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent, 2008. Manuscript available from `http://www-stat.stanford.edu/~hastie/Papers/glmnet.pdf`.

X. Gao, D.Q. Pu, Y. Wu and H. Xu. Tuning parameter selection for penalized likelihood estimation of inverse covariance matrix, 2009. *Preprint.* Available at `http://arxiv.org/PS_cache/arxiv/pdf/0909/0909.0934v1.pdf`

A. Genkin, D. Lewis and D. Madigan. Bayesian Logistic Regression for text categorization. *Technometrics* **49,** 291–304, 2007.

H. Höfling and R. Tibshirani. Estimation of Sparse Binary Pairwise Markov Networks Using Pseudo-likelihoods. *Journal of Machine Learning Research* **10,** 883–906, 2009.

K. Koh, S. Kim and S. Boyd. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research* **8,** 1519–1555, 2007.

S. Lee, V. Ganapathi and D. Koller. Efficient structure learning of Markov networks using L1- regularization. *Advances in Neural Information Processing Systems 19*, 817–824, 2007.

S. Lee, H. Lee, P. Abeel and A. Ng. Efficient ℓ_1-regularized logistic regression. *Proceeding of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 2006.

P. Mccullagh and J.A. Nelder. *Generalized Linear Models, Second Edition.* London: Chapman and Hall, 1989.

N. Meinshausen and P. Bühlmann. High-dimensional graphs with the lasso. *Annals of Statistics* **34,** 1436–1462, 2006.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, B* **58,** 267–288, 1996.

M.J. Wainwright and M.I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing* **54,** 2099–2109, 2006.

M.J. Wainwright, P. Ravikumar and J.D. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. *Proceedings of Advances in Neural Information Processing Systems* 1465–1472, 2006.

P. Wang, D.L. Chao and L. Hsu. Learning networks from high dimensional binary data: An application to genomic instability data. *Biometrics*, 2010.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics.* Chichester: John Wiley and Sons, 2000.

World Heath Organization. *International classification of diseases,* 10th revision. Geneva: WHO, 1994.

M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika* **94,** 19–35, 2007.